Metodat për Zgjidhjen e Sistemeve të Ekuacioneve Lineare: Metoda e Jacobi-t dhe Metoda Gauss–Seidel

1 Hyrje

Qëllimi i këtij materiali është të paraqesë metodat iterative për zgjidhjen e sistemeve të ekuacioneve lineare të formës

$$Ax = b, \tag{1}$$

ku A është një matricë $n \times n$, x është vektori i panjohur dhe b është vektori i anës së djathtë. Metodat më të përdorura iterative janë:

- Metoda e Jacobi-t
- Metoda Gauss–Seidel

Këto metoda përdoren zakonisht kur matrica është e rrallë, e madhe, ose kur zgjidhjet direkte (si eliminimi i Gaussit) janë të kushtueshme.

2 Metoda e Jacobi-t

Le të ndajmë matricën A në:

$$A = D + R, \tag{2}$$

ku D është diagonalia e matricës, ndërsa R përmban të gjithë elementët jashtë diagonalës. Formula e iteracionit Jacobi është:

$$x^{(k+1)} = D^{-1}(b - Rx^{(k)}). \tag{3}$$

Në MATLAB, implementimi i thjeshtë është:

```
--------------------------------------------------------------
function [x, k, res_hist] = metoda_jacob(A, b, x0, tol, maxIter)
n = size(A,1)
D = diag(diag(A))
R = A - D
Dinv = diag(1./diag(D))
x = x0
res_hist = zeros(maxIter,1)
for k = 1:maxIter
    x_new = Dinv * (b - R * x)
    res_hist(k) = norm(A*x_new - b)
    if norm(x_new - x) < tol
```

```
        x = x_new
        res_hist = res_hist(1:k);
        return
    end
    x = x_new
end
res_hist = res_hist(1:maxIter)
end
--------------------------------------------------------------
```

3 Metoda Gauss–Seidel

Metoda Gauss–Seidel është e ngjashme me Jacobi-n, por përdor vlerat e reja të llogaritura menjëherë. Formula është:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)} \right). \tag{4}$$

Implementimi në MATLAB:

```
--------------------------------------------------------------
function [x, k, res_hist] = metoda_gauss_seidel(A, b, x0, tol, maxIter)
n = length(b)
x = x0
res_hist = zeros(maxIter,1)
for k = 1:maxIter
    x_old = x
    for i = 1:n
        s1 = A(i,1:i-1) * x(1:i-1)
        s2 = A(i,i+1:n) * x_old(i+1:n)
        x(i) = (b(i) - s1 - s2) / A(i,i)
    end
    res_hist(k) = norm(A*x - b)
    if norm(x - x_old) < tol
        res_hist = res_hist(1:k);
        return
    end
end
res_hist = res_hist(1:maxIter)
end
--------------------------------------------------------------
```

4 Shembull Testimi

Le të zgjidhim sistemin:

$$\begin{cases} 10x_1 - x_2 + 2x_3 = 6 \\ -x_1 + 11x_2 - x_3 = 25 \\ 2x_1 - x_2 + 10x_3 = -11 \end{cases} \tag{5}$$

Ky sistem ka zgjidhjen e saktë:

$$x = \left(1;\ 2;\ -1\right) \tag{6}$$

Kodi testues në MATLAB:

```
A = [10 -1 2; -1 11 -1; 2 -1 10];
b = [6; 25; -11];
x0 = [0;0;0];
tol = 1e-6;
maxIter = 100;
[x_j, k_j, res_j] = metoda_jacob(A, b, x0, tol, maxIter)
[x_gs, k_gs, res_gs] = metoda_gauss_seidel(A, b, x0, tol,maxIter)
```

5 Përfundime

Metoda Gauss–Seidel zakonisht konvergon më shpejt se Jacobi, pasi përdor vlerat e reja menjëherë gjatë iterimit. Megjithatë, të dyja metodat funksionojnë mirë kur matrica A është rreptësisht diagonal-dominante.